REMARKS

This Amendment is in response to the Office Action mailed November 16, 2007. In the Office Action, the Examiner rejected claims 1-45 under 35 U.S.C. § 103. In this response, claims 1, 14, 21-41, 43 and 44 have been amended. No claims have been added or cancelled. Therefore, claims 1-45 are presented for examination. Reconsideration in light of the amendments and remarks made herein is respectfully requested.

Rejection Under 35 U.S.C. § 112

Claims 43 and 44 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which Applicant regards as the invention. In particular, the Examiner noted that the claims included a trademark/tradename. Applicant has accordingly amended the claims to recite "Java Virtual Machine" and have removed the trademark/tradename references. In light of the amendments, Applicant respectfully submits that the rejection has been overcome.

Rejection Under 35 U.S.C. § 101

Claims 21-40 are rejected under 35 U.S.C. § 101 as being directed to non-statutory subject matter. In particular the Examiner notes that the claims can reasonably be interpreted as computer program modules/software per se (Office Action, mailed November 16, 2007, page 6). The Examiner goes on to note, however, that claimed computer readable storage is a computer element and thus would be statutory. Applicant has accordingly amended claims 21-40 to recite a "computer-readable storage medium storing instructions for a Web application framework, which when executed by a computer system, causes the computer system to perform a method."

In light of the amendments, Applicant submits that claims 21-40 are directed to statutory subject matter under § 101, and respectfully requests withdrawal of the rejections.

Rejection Under 35 U.S.C. § 103

Claims 1-12, 15, 17-32, 35, 37-41, 43, and 44 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Publication No. 2002/0129060 of Rollins et al. (hereinafter "Rollins") in view of U.S. Patent No. 6,675,354 of Claussen et al. (hereinafter "Claussen"). Applicant respectfully disagrees.

Rollins describes a system and method for generating multiple customizable interfaces for XML documents (Rollins, Abstract). A component generation engine uses XML schema, which defines the structure of an XML document, and optional user preferences to define multiple components (Rollins, page 2, paragraph 0031; page 3, paragraphs 36-37). Then, a code-generator generates "a set of Java classes designed to mediate communication between the user and the synchronized tree manager" (Rollins, page 3, paragraph 0038).

Claussen describes processing and handling custom tags in a document (Claussen, Column 3, lines 14-29). The custom tags may be processed regardless of the type of tag that is encountered (Claussen, Column 30, lines 30-42). To avoid errors in processing the various tags, a tag preprocessing technique is utilized to classify the tag according to a tag library (Claussen, Column 6, lines 8-32). However, Claussen merely describes routines for translating various custom tags.

Claim 1, as amended, recites:

> In a computer system, an improved method for developing a Web application, the method comprising:
> providing a Web application development framework, said framework including an abstract command tag that predefines at least some generic Web application activities;

specifying at least one custom action that is desired to be performed by a Web application;

creating an object-oriented programming language (OOPL) class that extends the abstract command tag for providing execution logic for said at least one custom action, in addition to pre-existing logic that supports said at least some generic Web application activities, thereby creating a corresponding customized command tag that is capable of being embedded within a Web page;

embedding the customized command tag in a Web page of the Web application; and

upon execution of the Web application including an embedded customized command tag in a Web page, invoking the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions of the Web application and run-time values for one or more attributes included in the customized command tag.

(Emphasis Added)

Applicant respectfully submits that Rollins and Claussen, taken alone or in combination, fail to describe or suggest a method for developing a Web application that includes "upon execution of the Web application including an embedded customized command tag in a Web page, invoking the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions of the Web application and run-time values for one or more attributes included in the customized command tag."

Rollins describes a component generation engine that interprets an XML document based upon user preferences and various modes selected for the XML document. Then, the XML document can be presented from the components generated by the component generation engine (Rollins, paragraphs [0031] and [0036-0037]). Although Rollins briefly mentions that web programming languages utilize tags (*See* Rollins, paragraphs [0004-0010]), Rollins is completely silent as to the embedding and conditional execution of customized tags in a web application, as recited in claim 1, as amended. The Examiner acknowledges that Rollins fails to teach or suggest these features (Office Action, mailed November 16, 2007, page 8).

The Examiner therefore relies on Claussen as teaching the conditional execution of customized tags (Office Action, mailed November 16, 2007, pages 8-9). Applicant respectfully disagrees.

Claussen describes a method for pre-processing tags in a document based on whether or not those tags are present in a tag library and/or whether the tags are case sensitive tags. The pre-processing system of Claussen then rewrites tags in the document, as necessary, so that the document is compatible with a DOM format and the tag may be supplied to a tag handler (Claussen, column 3, lines 14-29; column 6, lines 8-32). Thus, Claussen describes rewriting tags, as necessary, to comply with tag library descriptors and case sensitivity for a DOM document. Claim 1 as amended, however, recites "upon execution of the Web application including an embedded customized command tag in a Web page, invoking the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions of the Web application and run-time values for one or more attributes included in the customized command tag." That is, the custom action is conditionally executed based on runtime conditions of the Web application and attributes in the tag. Because Claussen teaches rewriting tags to comply with tag library descriptors and case sensitivity for a DOM document, Claussen may at best teach conditionally rewriting tag names based on the current tag name, but fails describe any conditional execution of tags based on status of a Web application or on the attributes in a tag.

Furthermore, in the passage cited by the Examiner, Claussen recites:

> In an illustrative embodiment, a document object model (DOM) tree is processed to identify custom tags. Upon encountering a custom tag, an appropriate tag handler (e.g., a Java object, an XSL stylesheet, or the like) is invoked. According to the invention, a tag registration routine is used for recognizing and handling case-insensitive custom tags. As a servlet engine is examining a tag name, if the name does not match one of the registered tags, the routine converts the name to lower or neutral case. If the tag recognition routine recognizes the name as one of

> the case-insensitive tags, it converts the attributes to the appropriate case, and hands the resulting element off to a correct tag handler for processing.
>
> (Claussen, column 3, lines 30-42)

In the above passage, Claussen merely describes that tags in a document are analyzed, converted to a proper case if recognized, converted to a neutral case if not recognized, and the results handed to a tag handler for processing. Processing all tags and rewriting tag cases, however, fails to teach or even suggest "invoking the customized command tag for conditionally executing said specified at least one custom action." In Claussen, all tags are analyzed and processed, even though some tags names are rewritten. Thus, Claussen fails to describe conditionally executing custom any actions of an invoked tag, and merely describes rewriting tag names while still processing all tags. There is no hint or suggestion in Claussen that the actions specified in a customized command tag are conditionally executed, let alone conditionally executed "based on run-time conditions and run-time values for one or more attributes included in the customized command tag."

Thus, Rollins and Claussen, whether taken alone or in combination, fail to teach or suggest "upon execution of the Web application including an embedded customized command tag in a Web page, invoking the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions of the Web application and run-time values for one or more attributes included in the customized command tag." Therefore, Applicant respectfully submits that claim 1, and thus dependent claims 2-12, 15, 17-20, are not rendered obvious by Rollins and Claussen.

Claim 21, as amended, recites:

> A computer-readable storage medium storing instructions for a Web application framework, which when executed by a computer system, causes the computer system to perform a method comprising:
>
> specifying an abstract command tag that predefines at least some generic Web application activities;
>
> providing a programming environment for:
>
> (i) specifying at least one custom action that is desired to be performed by a Web application under development, by supporting creation of an object-oriented programming language (OOPL) class that extends the abstract command tag for providing execution logic for said at least one custom action, thereby creating a corresponding customized command tag that is capable of being embedded within a Web page, wherein said customized command tag includes the ability to conditionally execute said specified at least one custom action based on run-time conditions; and
>
> (ii) enabling embedding of the customized command tag in a Web page of the Web application;
>
> wherein execution of the Web application includes invoking the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions of the Web application and run-time values for one or more attributes included in the customized command tag.

(Emphasis Added)

As discussed above, with respect to claim 1, Rollins and Claussen fail to teach or suggest embedding and conditional executing customized tags in a web application based on run-time conditions of the Web application and run-time values for one or more attributes included in the customized command tag. Claim 21 as amended recites "(ii) enabling embedding of the customized command tag in a Web page of the Web application; wherein execution of the Web application includes invoking the customized command tag for conditionally executing said specified at least one custom action based on run-time conditions of the Web application and run-time values for one or more attributes included in the customized command tag." Thus, claim 21, and the claims that depend form it, are not rendered obvious over Rollins in view of Claussen.

Claim 41, as amended, recites:

> An improved method for Web application development, the method comprising:
>
> providing a Web-based application development framework built from a set of object-oriented programming language (OOPL) classes, which extends an abstract command tag, said framework providing:
>
> a non-programmatic tag framework that implements the functionality of the application framework when executing within a Web page using dynamic scripting capability;
>
> tag-based Web application objects controlling program flow, executing user commands, representing application business objects, and constructing output;
>
> a non-programmatic tag framework that accesses data for logical business objects and allows page designers to specify an action to be performed;
>
> enabling the embedding of the tag-based Web application objects in a Web page of a Web application; and
>
> execution of the Web application including invoking the tag-based Web application objects for conditionally executing actions specified by page designers based on run-time conditions of the tag-based Web application and tag attributes in the tag-based Web application objects.

(Emphasis Added)

As discussed above, with respect to claims 1 and 21, Rollins and Claussen fail to teach or suggest embedding and conditional executing customized tags in a web application based on run-time conditions of the Web application and run-time values for one or more attributes included in the customized command tag. Claim as amended 41 recites "enabling the embedding of the tag-based Web application objects in a Web page of a Web application; and execution of the Web application including invoking the tag-based Web application objects for conditionally executing actions specified by page designers based on run-time conditions of the tag-based Web application and tag attributes in the tag-based Web application objects." Thus, claim 41, and the claims that depend form it, are not rendered obvious over Rollins in view of Claussen.

Therefore, Applicants respectfully request withdrawal of the rejection of claims 1-12, 15, 17-32, 35, 37-41, 43, and 44 under 35 U.S.C. §103(a) over Rollins in view of Claussen.

Claims 13, 14, 16, 33, 34, 36, 42, and 45 were rejected as being unpatentable over Rollins and Claussen in view of U.S. Patent No. 6,760,748 of Hakim (hereinafter "Hakim"). Applicants respectfully disagree.

As discussed above Rollins and Claussen fail to describe each and every limitation claimed by the Applicant. Hakim merely describes an electronic classroom in which data is transmitted from a teacher's terminal to student terminals (Hakim, Column 3, lines 15-21; Figure 2). Hakim does not address the use of customized tags in a Web application. Therefore, Hakim fails to remedy the shortcomings of Rollins and Claussen discussed above. Thus, Rollins, Claussen, and Hakim, alone or in combination, fail to render claims 1, 21, and 41, and thus dependent claims 13, 14, 16, 33, 34, 36, 42, and 45, obvious.

Claim 14 is not obvious over the references for the same reason as claim 1, on which it depends. Furthermore, claim 14 as amended recites "wherein said generic Web application activities include filtering of requests to match run-time attributes of the requests with the run-time values for the one or more attributes included in the customized command tag." The Examiner relies on Hakim for this limitation (Office Action, mailed November 16, 2007, pages 16-17). In the passage of Hakim relied upon by the Examiner, Hakim states "[w]ith the addition of optional components (plug-ins), it is possible to extend their functionality to perform detailed content filtering, report generation, virus scanning, and more" (Hakim, column 29, lines 40-43). As described in Hakim, a web browser may include plug-in applications so that the plug-ins can supplement existing proxy servers with key-word content filtering (Hakim, column 29, lines 48-57). A keyword filter plug-in, however, merely analyzes the text in a document to avoid supplying unwanted content to a web browser, but fails to describe or even suggest "filtering of requests to match run-time attributes of the requests with the run-time values for the one or more

attributes included in the customized command tag." In view of this, Applicant respectfully submits that claim 14 is not rendered obvious by a combination of Rollins, Claussen, or Hakim.

Conclusion

Applicant reserves all rights with respect to the applicability of the doctrine of equivalents. Applicant respectfully requests that a timely Notice of Allowance be issued in this case. If a telephone interview would expedite the prosecution of this application, the Examiner is invited to contact William L. Jaffe at (714) 557-3800.

Respectfully submitted,

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

Dated: April 16, 2008

By _____
William W. Schaal
Reg. No. 39,018
Tel.: (714) 557-3800 (Pacific Coast)